

Figure 1

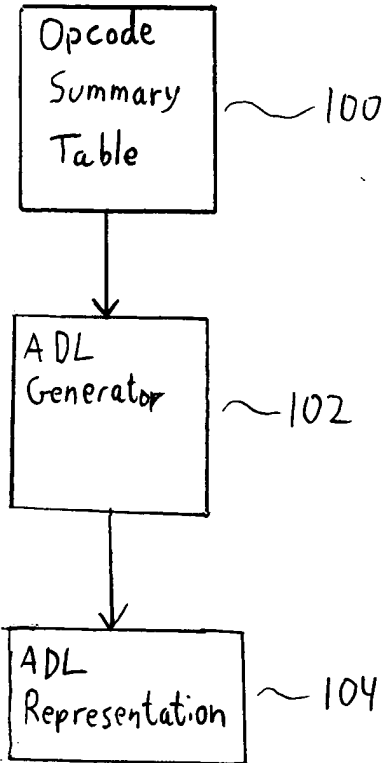


Figure 2

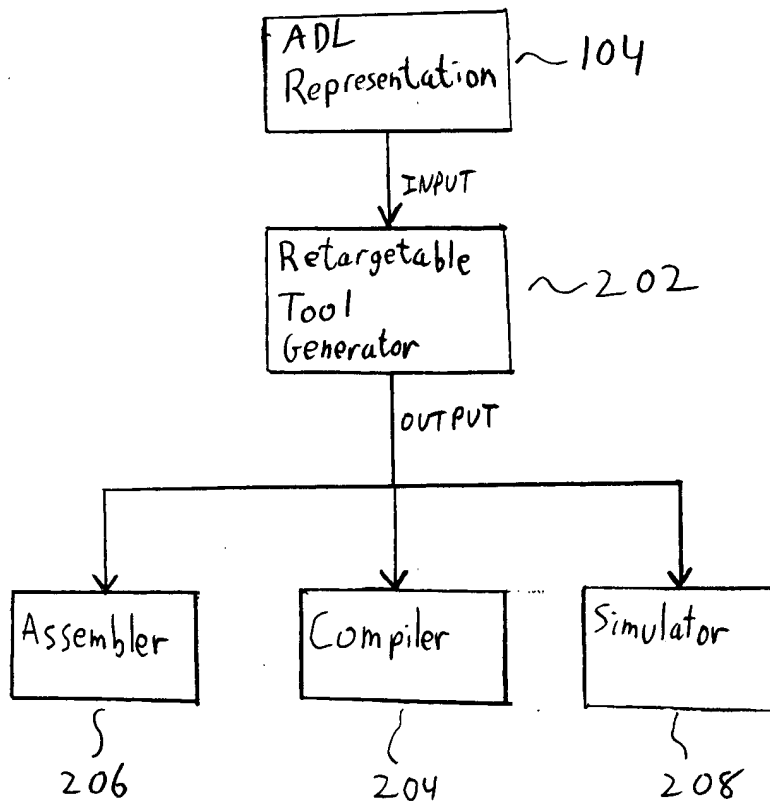


Figure 3

```

operation Ld_St_Imm_group
{ // The Load-Store Halfword Immediate Group
composition
{
    // select one of the four the instructions.
    inst = (LDRH || LDRSB || LDRSH || STRH)
    // and include their operand fields.
    && cond && P && U && W && Rn && Rd
    && immed // immed is a joining of immedH and immedL.
}
coding
{ // concatenate the opcode and operand fields
    self = cond:4 ## 0:3 ## P:1 ## U:1
    ## 1:1 ## W:1
    ## inst[2:2]:1 // the bit after the "W" operand.
    ## Rn ## Rd
    ## immed[4:7]:4 // immedH field.
    ## 1:1
    ## inst[1:0]:2 // the 2 bits between immedH & L.
    ## 1:1
    ## immed[0:3]:4 // immedL field.
}
}

```

Figure 4

```

operation Ld_St_half_group
{
    composition
    {
        // select one of the four the instructions.
        grp = (Ld_St_Imm_group || Ld_St_Reg_group)
        // and include their operand fields.
        && cond && P && U && W && Rn && Rd
    }
    coding
    { // concatenate the opcode and operand fields
        self = cond:4 ## 0:3 ## P:1 ## U:1
            ## grp[11:11]:1 // 1st "*", the immed/reg indicator bit.
            ## W:1
            ## grp[10:10]:1 // 2nd "*", the first inst opcode bit.
            ## Rn ## Rd
            ## grp[9:6]:4 // 3rd "*", the immedH field, or 0's.
            ## 1:1
            ## grp[5:4]:2 // 4th "*", the other inst opcode bits.
            ## 1:1
            ## grp[3:0]:4 // 5th "*", the immedL or Rm field.
    }
}

```

Figure 5

```

operation Ld_St_Imm_group
{
    composition
    {
        // select one of the four the instructions.
        inst = (LDRH || LDRSB || LDRSH || STRH)
        // and include their operand fields.
        && immmed // immmed is a joining of immmedH and immmedL.
    }
    coding
    { // blindly concatenate the opcode and operand fields
        self = 1:1 // imm indicator, opcode bit before "W" field.
            ## inst[2:2]:1 // the bit after the "W" operand.
            ## immmed[4:7]:4 // immmedH field.
            ## inst[1:0]:2 // the 2 bits between immmedH & L.
            ## immmed[0:3]:4 // immmedL field.
    }
}

```

Figure 6

```

operation Ld_St_Reg_group
{
    composition
    {
        // select one of the four the instructions.
        inst = (LDRH || LDRSB || LDRSH || STRH)
        // and include their operand fields.
        && Rm // the extra register operand.
    }
    coding
    { // blindly concatenate the opcode and operand fields
        self = 0:1 // reg indicator, opcode bit before "W" field.
            ## inst[2:2]:1 // the bit after the "W" operand.
            ## 0:4 // equivalent to immedH field.
            ## inst[1:0]:2 // the 2 bits between immedH & L.
            ## Rm:4
    }
}

```



Figure 8

